

Specification and Analysis of Weakly Hard Real-Time Systems

Ph.D. Thesis. Guillem Bernat
Departament de Ciències Matemàtiques i Informàtica.
Universitat de les Illes Balears.
Supervisor: Albert Llamósí

January 1998

Abstract

A real-time system is one in which the temporal aspects of its behaviour are part of their specification. The correctness of the system depends not only on the logical results of the computation, but also on the time at which the results are produced. This is due to the fact that these systems are used to interact with their environment, which also changes in time, as it is the case in applications related to automated control, manufacturing, robotics, avionics, telecommunication, multimedia, virtual reality, etc.

In all these domains, the control programs are not intended to terminate, but to loop endlessly examining input data and reacting accordingly. Therefore, the specification of real-time systems relies on this cyclic nature and is expressed in terms of operations —also called tasks— that have to be performed periodically and finish within a given deadline.

Traditionally, real-time systems are classified as being either *hard* or *soft*. For hard real-time systems, it is imperative that no deadline be missed, whilst in soft real-time systems it is acceptable to miss some of them *occasionally*. In practical engineering contexts, the occasional loss of some deadline can be tolerated. This is either because the consequences of the loss can be negligible (*i.e.* one defectuous part per thousand) or because the robustness of the involved control algorithms imply the ability to react properly at the next invocation step without serious consequences.

Nevertheless, the term *occasional* is so ambiguous that it has no practical meaning for a specification. For systems that may tolerate some degree of missed deadlines, the way that these missed and met deadlines are distributed is important. For instance, audio and video systems are very sensitive to the consecutiveness of the missed deadlines because a missed deadline corresponds to a gap in the output signal, and if these gaps occur consecutively, the quality

of the output is lower than if those gaps are non-consecutive.

Not having to meet *every* deadline allows the size of the system's resources to be smaller than it would be for meeting all of them. This permits the creation of simpler and more cost-effective systems that make better use of the available resources whilst guaranteeing a reasonably good level of service.

The first main goal achieved by this thesis is to provide an appropriate conceptual framework for specifying real-time systems that can tolerate occasional losses of deadlines. For this purpose, the concept of hard real-time system is weakened to include systems for which the distribution of its met and lost deadlines is precisely bounded. Systems that must meet all their deadlines are a particular case of our definition and will be referred to as *strongly hard real-time systems*, whenever the distinction is relevant. Hard systems that are not strongly hard will be called *weakly hard real-time systems* or simply weakly hard systems.

This tolerance is established by specifying, for each task, the bounds for missed deadlines allowed in every window of m consecutive invocations. This allows several kinds of restrictions to be introduced, $\binom{n}{m}$ being the simplest one. $\binom{n}{m}$ means that the task must meet n deadlines in any window of m consecutive invocations. The thesis analyses the properties of this and other kinds of temporal restrictions, as well as the possibilities of combining them.

The second main goal of the thesis is to provide a technique to analyse whether an implementation of the system will meet the specification. The implementation of real-time systems can be basically done in three different ways, or by a combination of them. Namely, cyclic executives, concurrent task systems activated by events and data flow systems.

Cyclic executives are easy to analyse but their construction and maintenance is hard because it involves solving NP-complete problems. Concurrent task systems are harder to analyse because their behaviour is nondeterministic and therefore they are less predictable. In particular, they are very sensitive to the scheduling algorithms and priority assignments to the tasks. Data flow or message passing systems are even harder to analyse.

From the point of view of implementation, the work carried out by this thesis addresses only systems of the second type. A lot of research has been done on schedulability analysis although it has always been restricted to strongly hard systems. This thesis has extended their results and analysis techniques to weakly hard systems in the context of fixed priority scheduling of monoprocessor systems.

The third main goal of the thesis is to provide design guidelines for weakly hard systems. One of the interesting results is that, for weakly hard systems, the deadline monotonic priority ordering of tasks is not optimal. A technique to compute the optimal priority ordering is provided.

This thesis also provides two additional contributions to the design of weakly hard systems. Namely, a scheduling algorithm for effectively managing systems made up by both hard and soft tasks, and a unified approach for the design of weakly hard and fault tolerant systems.

A frequent problem is to integrate hard tasks with soft tasks. Soft tasks have no deadline associated with them, otherwise they would be weakly hard according to our terminology. For soft tasks the design goal is to minimize their response times provided that the hard tasks satisfy their specifications. Several algorithms have been proposed to solve this problem in the context of strongly hard tasks. This thesis presents a simple and effective scheduling mechanism based on dual priority scheduling for systems having weakly hard tasks. With a very low run-time overhead, the mechanism exploits the fact that hard tasks are weakly hard in order to provide better response times for soft tasks than current approaches do.

Missing a deadline is a fault in a real-time system. So the concept of fault-tolerance can be applied to real-time systems under this point of view. The thesis presents a fault-tolerant real-time architecture, the *Redundancy eXecutive* (RX), and a scheduling paradigm based on weakly hard tasks. The proposed approach uses the redundancy introduced for guaranteeing the fault tolerance in the classical sense to also reduce the amount of lost deadlines. Therefore, a set of redundant weakly hard systems make up a less weakly hard system. Schedulability analysis tests are developed for predicting the temporal behaviour of a replicated real-time system based on this architecture by the introduction of the concept of dual time-outs to mask both data and timing errors. Its application leads to achieving cost-effective fault-tolerance because the resources for each of the component systems can be adjusted to a smaller size.